A JDBC MIDDLEWARE FOR DATA REPLICATION

This dissertation was submitted to the
Department of Computer Science and Engineering
Of the
University of Moratuwa
In partial fulfilment of the requirements for the
Degree of M.Sc. in Computer Science

By
Aruna Dissanayake
Department of Computer Science and Engineering
University of Moratuwa, Sri Lanka
December 2008

Abstract

Data is an important part of today's IT infrastructure: both companies and state institutions rely on database systems to store most of their important data. As companies move toward systems that are distributed, research and development in data-replication is becoming increasingly important. In simple terms, replication is a form of distributing copies of data to other machines Data replication is a widely used approach for providing high availability, fault tolerance and better performance (w.r.t reads). With the data being redundantly available from more than one source user can be shielded from network failures, delays and data source failures. One of the major goals in replication is maintaining data consistency amongst the various replicas.

However achieving consistency is a complex problem and it can adversely undermine its advantages. Therefore building an efficient, consistent replicated database is still an open research topic, though both the database community and the distributed systems community have been utilizing replication for nearly three decades.

Moreover, if two updates happen concurrently, it is often also required that each copy is updated in the same order. This is very expensive in practice and can degrade performance of the end application. Hence the choic^vofthe consistency model mostly depends on the application using the replication scheme. For instance mission critical applications need high consistency while

mobile applications are willing to trade off consistency, for availability and performance.

Most of the researches have therefore focused on a specific application domain and interns implement a particular consistency model, which does not suit everyone. Moreover both the industry and research community seem to favour relaxed consistency models like sequential consistency, because higher consistent models are more restrictive.

In this research we propose a novel replication scheme that can be tuned to match varying degrees of consistency requirements, of a wider user community. We implement this replication scheme as a JDBC middleware. Our JDBC implementation can replicate update/insert requests across a large number of nodes (data sources/replicas). The JDBC interface hides the complexity of distribution and replication while providing a consistent and transparent view of the data store to the application developers.